

IN THE CLAIMS:

Please rewrite the claims to read as follows:

- 1 1. (ORIGINAL) A method for converting a file access data structure from a first en-
2 dianness to a second endianness, the method comprising the steps of:
3 identifying, from a descriptor look up table, a series of actions to perform on ele-
4 ments of the file access data structure; and
5 performing the identified series of actions on the elements of the file access data
6 structure.
- 1 2. (ORIGINAL) A method of converting elements of a file access data structure from a
2 first endianness to a second endianness, the method comprising the steps of:
3 determining if the file access data structure is a critical path data structure;
4 converting, in response to the file access data structure being a critical path data
5 structure, the elements from the first endianness to the second endianness using a set of
6 specific code functions;

7 converting, in response to the file access data structure not being a critical path
8 data structure, a header of the file access data structure from the first endianness to the
9 second endianness using a second set of specific code functions; and
10 calling a byte swapping engine to convert selected elements of the file access data
11 structure from the first byte order to the second byte order.

1 3. (ORIGINAL) The method of claim 2 wherein the file access data structure further
2 comprises a direct access file access data structure.

1 4. (ORIGINAL) A system for converting elements of a file access data structure from a
2 first endianness to a second endianness, the system comprising:

3 an input buffer, the input buffer storing the file access data structure to be con-
4 verted;

5 a byte swapping engine, the byte swapping engine operative interconnected with a
6 descriptor table; and

7 an output buffer, the byte swapping engine placing the file access data structure in
8 the output buffer after conversion.

1 5. (ORIGINAL) The system of claim 4 wherein the descriptor table further comprises a
2 set of entries describing various file access data structures, each entry further comprising
3 a size field and an operation field.

1 6. (ORIGINAL) The system of claim 4 wherein the file access data structure further
2 comprises a direct access file access data structure.

1 7. (ORIGINAL) A method for converting a data structure from a first byte order to a
2 second byte order, the method comprising the steps of:
3 reading an element entry from a descriptor table;
4 performing an action on an element of the data structure, the action being defined
5 in the element entry read from the descriptor table; and
6 placing the element in an output buffer.

1 8. (ORIGINAL) The method of claim 7 wherein the step of performing an action on an
2 element further comprises the step of copying the element from an input buffer to the
3 output buffer.

1 9. (ORIGINAL) The method of claim 7 wherein the step of performing an action on an
2 element further comprises the step of byte swapping the element.

1 10. (ORIGINAL) The method of claim 7 wherein the element entry of the descriptor
2 table further comprises a field describing a size of the element and a field describing an
3 action to be performed.

1 11. (ORIGINAL) A file server for use in a network storage environment, the file server
2 comprising:

3 a byte swapping engine, the byte swapping engine performing a defined operation
4 on each of a plurality of elements of a file access data structure.

1 12. (ORIGINAL) The file server of claim 11 wherein the file server further com-
2 prises a descriptor look up table, the descriptor look up table having a plurality of entries,
3 each of the plurality of entries associated with a specific file access data structure.

1 13. (ORIGINAL) The file server of claim 12 wherein each of the plurality of entries
2 further comprises a plurality of elements, each of the elements having a size field and an
3 operation field.

1 14. (ORIGINAL) The file server of claim 13 wherein the defined operation is de-
2 fined by the operation field of the entry associated with the file access data structure.

1 15. (ORIGINAL) A computer-readable medium, including program instructions execut-
2 ing on a computer, for converting elements of a file access data structure from a first en-
3 dianness to a second endianness, the method comprising the steps of:
4 determining if the file access data structure is a critical path data structure;

5 converting, in response to the file access data structure being a critical path data
6 structure, the elements from the first endianness to the second endianness using a set of
7 specific code functions;

8 converting, in response to the file access data structure not being a critical path
9 data structure, a header of the file access data structure from the first endianness to the
10 second endianness using a second set of specific code functions; and

11 calling a byte swapping engine to convert selected elements of the file access data
12 structure from the first byte order to the second byte order.

1 16. (ORIGINAL) A method for converting elements of a file access data structure from a
2 first endianness to a second endianness, the method comprising the steps of:

3 determining a type of the file access data structure;

4 processing, in response to the file access data structure of being of a first type, the
5 file access data structure along a first processing path;

6 processing, in response to the file access data structure being of a second type, the
7 file access data structure along a second processing path.

1 17. (ORIGINAL) The method of claim 16 wherein the first type further comprises a
2 critical path data structure.

1 18. (ORIGINAL) The method of claim 16 wherein the first processing path further com-
2 prises a set of specifically coded functions.

1 19. (ORIGINAL) The method of claim 16 wherein the second processing path further
2 comprises a byte swapping engine.

1 20. (PREVIOUSLY PRESENTED) A method for converting a data structure, compris-
2 ing:

3 calling a byte-swapping engine;

4 providing a file access data structure as input to the byte-swapping engine;

5 providing a descriptor look up table to the byte-swapping engine;

6 identifying, from the descriptor look up table, a series of actions to perform on
7 elements of the file access data structure in order to swap bytes of the file access data
8 structure from a first endianness to a second endianness; and

9 performing the identified series of actions on the elements of the file access data
10 structure.

1 21. (PREVIOUSLY PRESENTED) The method as in claim 20, further comprising:

2 using as the file access data structure a file having Direct Access File System
3 (DAFS) protocol.

1 22. (PREVIOUSLY PRESENTED) The method as in claim 20, further comprising:

2 determining if the file access data structure is a critical path data structure, and if
3 it is, perform byte swap operations using specific code functions.

1 23. (PREVIOUSLY PRESENTED) The method as in claim 20, further comprising:
2 determining if the file access data structure is a critical path data structure, and if
3 it is not, perform byte swap operations on a data structure header.

1 24. (PREVIOUSLY PRESENTED) The method as in claim 20, further comprising:
2 swapping bytes of the data structure as needed, in response to swapping bytes of
3 the file access data structure.

1 25. (PREVIOUSLY PRESENTED) The method as in claim 20, further comprising:
2 determining if an element entry of the descriptor look up table is nested;
3 branching to the nested entry;
4 identifying, from the descriptor look up table, a series of actions to perform on
5 elements of the nested entry in order to swap bytes of the entry from a first endianness to
6 a second endianness.

1 26. (PREVIOUSLY PRESENTED) A computer to convert a data structure, comprising:
2 means for calling a byte-swapping engine;
3 means for providing a file access data structure as input to the byte-swapping en-
4 gine;
5 means for providing a descriptor look up table to the byte-swapping engine;

6 means for identifying, from the descriptor look up table, a series of actions to per-
7 form on elements of the file access data structure in order to swap bytes of the file access
8 data structure from a first endianness to a second endianness; and
9 means for performing the identified series of actions on the elements of the file
10 access data structure.

1 27. (PREVIOUSLY PRESENTED) The computer as in claim 26, further comprising:
2 means for using as the file access data structure a file having Direct Access File
3 System (DAFS) protocol.

1 28. (PREVIOUSLY PRESENTED) The computer as in claim 26, further comprising:
2 means for determining if the file access data structure is a critical path data struc-
3 ture, and if it is, perform byte swap operations using specific code functions.

1 29. (PREVIOUSLY PRESENTED) The computer as in claim 26, further comprising:
2 means for determining if the file access data structure is a critical path data struc-
3 ture, and if it is not, perform byte swap operations on a data structure header.

1 30. (PREVIOUSLY PRESENTED) The computer as in claim 26, further comprising:
2 means for swapping bytes of the data structure as needed, in response to swapping
3 bytes of the file access data structure.

1 31. (PREVIOUSLY PRESENTED) The computer as in claim 26, further comprising:
2 means for determining if an element entry of the descriptor look up table is
3 nested;
4 means for branching to the nested entry;
5 means for identifying, from the descriptor look up table, a series of actions to per-
6 form on elements of the nested entry in order to swap bytes of the entry from a first en-
7 dianness to a second endianness.

1 32. (PREVIOUSLY PRESENTED) A computer readable media, comprising:
2 said computer readable media containing instructions for execution on a processor
3 for the practice of a method for converting a data structure, the method having the steps
4 of,
5 calling a byte-swapping engine;
6 providing a file access data structure as input to the byte-swapping engine;
7 providing a descriptor look up table to the byte-swapping engine;
8 identifying, from the descriptor look up table, a series of actions to perform on
9 elements of the file access data structure in order to swap bytes of the file access data
10 structure from a first endianness to a second endianness; and
11 performing the identified series of actions on the elements of the file access data
12 structure.

1 33. (PREVIOUSLY PRESENTED) Electromagnetic signals propagating on a computer
2 network, comprising:
3 said electromagnetic signals carrying instructions for execution on a processor for
4 the practice of a method for converting a data structure, the method having the steps of,
5 calling a byte-swapping engine;
6 providing a file access data structure as input to the byte-swapping engine;
7 providing a descriptor look up table to the byte-swapping engine;
8 identifying, from the descriptor look up table, a series of actions to perform on
9 elements of the file access data structure in order to swap bytes of the file access data
10 structure from a first endianness to a second endianness; and
11 performing the identified series of actions on the elements of the file access data
12 structure.